

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ



ЗАТВЕРДЖУЮ
ФІТ

Говорушенко Т.О.
1 вересня 2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Кросплатформне програмування

Назва

Галузь знань 12 – Інформаційні технології

Спеціальність 123 – Комп'ютерна інженерія очна денна форма здобуття освіти

Освітня програма Комп'ютерна інженерія та програмування

Статус дисципліни: вибіркова, дисципліна професійної підготовки

Факультет – Інформаційних технологій

Кафедра – Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Курс	Семестр	Загальне навантаження		Кількість годин						Форма семестрового контролю			
			Кредити ЄКТС	Години	Аудиторні заняття						Курсовий проект	Курсова робота	Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття	Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС				
ОД		не парний	8	240	102	34	34	34		138			+	
Разом			8	240	102	34	34	34		138			1	

Програма складена Денисюком Д.О.
Підпис Ініціали, прізвище викладача(ів)

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем

Протокол 2 від 30.08.2024 р.

Зав. кафедри комп'ютерної інженерії та інформаційних систем Засорнова І.О.
Підпис Ініціали, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій № 1 від 05.09.2024р

Голова Вченої ради Говорушенко Т.О.
Підпис Ініціали, прізвище

Вступ

Мета викладання дисципліни. Дисципліна “Крос-платформене програмування” відноситься до циклу вибіркових дисциплін, забезпечує підготовку студентів спеціальності “Комп’ютерна інженерія” у напрямку розробки програмних додатків із використанням принципів об’єктно-орієнтованого програмування для різних операційних систем.

Предмет дисципліни. Принципи побудови крос-платформених додатків, мережеву взаємодію між ними та роботою із базами даних.

Завдання дисципліни.

- навчити застосовувати класи та ієрархії класів для побудови програмних додатків на мові Java.
- розглянути основні можливості побудови користувацьких інтерфейсів за допомогою візуальних компонентів Java Swing.
- навчити використовувати мережеві технології мови програмування Java для забезпечення зв’язку між програмними додатками.
- розглянути основні можливості мови програмування Java для створення багатопотокових додатків.

В результаті вивчення курсу «Крос-платформене програмування» студент повинен досягти таких результатів навчання (сукупність знань, умінь, навичок, компетентностей):

знати:

- особливості стандартних класів у мові Java та як використовувати їх основні методи; способи оголошення користувацьких класів та типи їх конструкторів; засоби обмеження доступу до класів та їх членів; методи створення користувацьких інтерфейсів у мові Java та особливості візуальних компонентів; загальні відомості про створення багатопотокових додатків та способів організації мережевої комунікації між ними;

вміти:

- використовувати оператори мови Java та ключові слова для побудови алгоритмів у рамках поставленої задачі;
- описувати та створювати класи та їх користувацькі конструктори, методи та поля;
- будувати ієрархію класів за допомогою наслідуваних класів для вирішення поставленої задачі із використанням принципів об’єктно-орієнтованого програмування;
- використовувати модифікатори доступу до класів та їх членів для забезпечення захисту даних у користувацьких програмах;
- працювати із візуальними компонентами графічної бібліотеки для побудови користувацьких інтерфейсів використовуючи один код для усіх настільних операційних систем;
- оперувати потоками в операційних системах для побудови програмних додатків із задіянням техніки паралельних обчислень;
- проводити первинне налаштування баз даних для зберігання даних використовуючи візуальні Java-додатки

бути здатним:

- до абстрактного мислення, аналізу і синтезу при побудові алгоритмів для розв'язування прикладних задач;
- до абстрактного мислення та аналізу при розробці ієрархій класів та описі користувацьких конструкторів та методів;
- до аналізу, пошуку та оволодінню сучасних практик та рекомендацій, що використовуються у розробці нового програмного забезпечення;
- розробляти, тестувати та створювати крос-платформені програмні додатки із візуальним користувацьким інтерфейсом для роботи із даними;
- аргументувати вибір методів розв'язування спеціалізованих прикладних задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

КРОС-ПЛАТФОРМЕНЕ ПРОГРАМУВАННЯ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	5
Кількість встановлених кредитів ЄКТС	8,0
Форми навчання, для яких викладається дисципліна	Очна денна

Результати навчання. ЯК ПРН Студент, який успішно завершив вивчення дисципліни, повинен: вміло застосовувати знання програмування та побудови алгоритмів, використовувати оператори мови програмування Java з метою реалізації крос-платформеного додатку для вирішення поставленої задачі; успішно використовувати набуті знання для опису, створення та використання класів, розробки ієрархії класів для реалізації ефективних додатків із використанням принципів та парадигм об'єктно-орієнтованого програмування; застосовувати модифікатори доступу при написанні програми для захисту даних та об'єктів в цілому від незапланованого втручання; використовувати візуальні компоненти для розробки користувацького інтерфейсу; застосовувати практики для побудови багатопотокових додатків, організувати мережеву комунікацію та налаштувати зв'язок із базами даних, для опрацювання даних;

Зміст навчальної дисципліни. Технологія Java та JVM. Стандартні класи та їх методи. Структури даних у Java. Колекції. Абстрактні класи та інтерфейси. Об'єктно-орієнтоване програмування. Поліморфізм. Наслідування. Інкапсуляція. Візуальні компоненти. Потoki. Технології мережевої комунікації у Java. Сокети. Робота із СКБД.

Запланована навчальна діяльність: лекції – 34 год., лабораторні заняття – 34 год., практичні заняття – 34 год., самостійна робота – 138 год.; разом – 240 год.

Методи навчання: словесні, наочні, проблемно-пошукові (лекції), частково-пошукові (лабораторні заняття) пояснювально-ілюстративні (практичні заняття), дослідницькі, частково-пошукові (самостійна робота).

Форми оцінювання результатів навчання: усне опитування, захист лабораторних та практичних робіт, тестовий контроль.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Core Java(TM), Volume I—Fundamentals, Cay S. Horstmann 2018p – 928p.
2. Head First. Java, Kathy Sierra, Bert Bates 2022p – 688p.
3. Thinking In Java 4th Edition, Bruce Eckel, 2022p – 1168p. .
4. DevOps Tools for Java Developers. Best Practices from Source Code to Production Containers, Stephen ChinMelissa McKayIxchel RuizBaruch Sadogursky 2022– 342p.
5. Java Cookbook: Problems and Solutions for Java Developers 4th Edition, Ian F. Darwin, 2020 – 600p.
6. Programming AWS Lambda: Build and Deploy Serverless Applications with Java 1st Edition, John ChapinMike Roberts, 2020 – 224p.
7. Herbert Schildt. Java. The complete Reference. Comprehensive Coverage of the Java Language. Eleventh Edition. McGraw-Hill Education (Publisher), 2019 – 1248p.
8. Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft. Modern Java in Action. Lambdas, streams and functional programming. Manning Publications, 2019 – 592p.

Викладачі: старший викладач Денисюк Дмитро Олександрович

1. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	лекції	лабораторні роботи	практичні роботи	самостійну роботу
Тема 1. Об'єктно-орієнтоване програмування у Java.	8	8	8	32
Тема 2. Опрацювання даних та їх відображення.	16	16	16	64
Тема 3. Додаткові бібліотеки Java.	10	10	10	42
Години	34	34	34	138
Разом за семестр:	240 год. / 8,0 кредитів			

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1 Зміст лекційного курсу

Номер лекції	Перелік змістових модулів, тем лекцій, їх анотації	Кількість годин
1.	Вступ до Java. Життєвий цикл. Робота JVM. Змінні. Типи даних. Приведення типів. Пріоритет операцій. Методи. Літ. [1,3,5]	2
2.	Робота із класами. Поняття класу та об'єкту. Методи створення об'єктів. Операції базового класу. Поля та методи об'єкту. Літ. [1,3,4,5]	2
3.	Об'єктно-орієнтоване програмування у Java. Поняття наслідування, поліморфізму та інкапсуляції. Приватні поля класів та методи доступу до них. Методи користувацьких класів. Літ. [1,3-5]	2
4.	Конструктори. Конструктори класів. Абстрактні класи. Інтерфейси. Вкладені класи. Клас Object та його методи. Літ. [1,3-5]	2
5.	Масиви. Одновимірні та багатовимірні масиви. Методи перебору масивів. Методи масивів. Поняття колекцій. Інтерфейс Collection. Літ. [1,3,5]	2
6.	Колекції. Типи колекцій. Клас ArrayList. Клас LinkedList. Клас HashSet. Клас Dictionary. Літ. [1,8]	2
7.	Потоки вводу та виводу. Потоки. Закриття потоків. Клас InputStream. Клас OutputStream. Форматований вивід. PrintStream та PrintWriter. Літ. [2,5]	2
8.	Робота із файлами. Читання та запис файлів. FileInputStream та FileOutputStream. Клас File. Робота із файлами та каталогами. Літ. [2,3,5]	2
9.	Введення у Java Swing. Графічні елементи. Компонування елементів. Типи панелей компонування. Літ. [1,3,5]	2
10.	Елементи управління інтерфейсів. Кнопки та мітки. Елементи: CheckBox, RadioButton. Текстові елементи. Події елементів. Літ. [1,5]	2
11.	Java та бази даних. Основи баз даних. Технологія JDBC. Огляд, встановлення та базові налаштування MySQL. Літ. [2,6]	2
12.	Робота із даними у MySQL. Підключення до бази даних. Додавання, видалення та зміна даних. Методи executeUpdate та executeQuery. Літ. [2,6]	2
13.	Мережеві Java додатки 1. Сокети. Огляд сокетів. Мережеві класи та інтерфейси Літ. [2,6-8]	2
14.	Мережеві Java додатки 2. Клас URLConnection. Організація клієнт-серверної взаємодії. Літ. [2,6-8]	2
15.	Потоки та процеси. Клас Thread. Створення та виконання потоку. Завершення та переривання потоку. Літ. [1,4,5,7]	2
16.	Багатопотокове програмування. Синхронізація потоків. Оператор synchronised. Семафори. Обмін даних між потоками. Літ. [1,4,5,7]	2
17.	Додаткові Java бібліотеки. Математичні обчислення та клас Math. BigInteger та BigDecimal. Робота із датами. Літ. [1,6]	2
Разом за семестр:		34

2.2 Зміст лабораторних занять

№ п/п	Теми лабораторних робіт		Кількість годин
1.	<i>Лабораторна робота №1</i>	Базові операції та класи у Java	2
2.	<i>Лабораторна робота №1</i>	Базові операції та класи у Java	2
3.	<i>Лабораторна робота №2</i>	Об'єктно-орієнтоване програмування у Java	2
4.	<i>Лабораторна робота №2</i>	Об'єктно-орієнтоване програмування у Java	2
5.	<i>Лабораторна робота №3</i>	Колекції та структури даних	2
6.	<i>Лабораторна робота №3</i>	Колекції та структури даних	2
7.	<i>Лабораторна робота №4</i>	Робота із файлами	2
8.	<i>Лабораторна робота №4</i>	Робота із файлами	2
9.	<i>Лабораторна робота №5</i>	Візуальні додатки та побудова користувацьких інтерфейсів	2
10.	<i>Лабораторна робота №5</i>	Візуальні додатки та побудова користувацьких інтерфейсів	2
11.	<i>Лабораторна робота №6</i>	Взаємодія із базами даних у Java	2
12.	<i>Лабораторна робота №6</i>	Взаємодія із базами даних у Java	2
13.	<i>Лабораторна робота №7</i>	Мережеві додатки	2
14.	<i>Лабораторна робота №7</i>	Мережеві додатки	2
15.	<i>Лабораторна робота №8</i>	Потоки та процеси	2
16.	<i>Лабораторна робота №8</i>	Потоки та процеси	2
17.	<i>Лабораторна робота №9</i>	Підсумкове заняття	2
Всього			34

2.3 Зміст практичних занять

№ п/п	Теми практичних робіт		Кількість годин
1.	<i>Практична робота №1</i>	Робота із змінними та методами. Типи даних.	2
2.	<i>Практична робота №2</i>	Створення класів та їх використання.	2
3.	<i>Практична робота №3</i>	Наслідування, ключові слова super та extends.	2
4.	<i>Практична робота №4</i>	Перевизначені методи.	2
5.	<i>Практична робота №5</i>	Масиви та робота методами масивів.	2
6.	<i>Практична робота №6</i>	Використання колекцій та їх методи.	2
7.	<i>Практична робота №7</i>	Потоки вводу та форматований вивід.	2
8.	<i>Практична робота №8</i>	Відкриття, модифікація та закриття файлів.	2
9.	<i>Практична робота №9</i>	Компонування графічних елементів.	2
10.	<i>Практична робота №10</i>	Елементи інтерфейсів та їх події.	2
11.	<i>Практична робота №11</i>	Технологія JDBC та налаштування MySQL.	2
12.	<i>Практична робота №12</i>	Базові операції при роботі із базою даних.	2
13.	<i>Практична робота №13</i>	Реалізація серверного та клієнтського додатків.	2
14.	<i>Практична робота №14</i>	Способи передачі даних по мережі.	2
15.	<i>Практична робота №15</i>	Потоки та інтерфейс Runnable	2
16.	<i>Практична робота №16</i>	Обмін даними між потоками. Семафори.	2
17.	<i>Практична робота №17</i>	Підсумкове заняття	2
Всього			34

2.4 Зміст самостійної (індивідуальної) роботи

Об'єм самостійної роботи з дисципліни "Крос-платформене програмування" становить 138 годин. Він включає опрацювання лекційного матеріалу, підготовку до виконання практичних та лабораторних робіт і їх захисту, підготовку до поточного контролю.

Керівництво самостійною роботою здійснює викладач згідно з розкладом консультацій в позаурочний час.

Номер тижня	Вид самостійної роботи	К-ть годин
1-2	Опрацювання лекційного матеріалу. Підготовка до ЛР1	16
3-4	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР1. Підготовка до ЛР2	16
5-6	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР2. Підготовка до ЛР3	16
7-8	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР3. Підготовка до ЛР4	16
9-10	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР4. Підготовка до ЛР5	16
11-12	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР5. Підготовка до ЛР6	16
13-14	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР6. Підготовка до ЛР7	16
15-16	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР7. Підготовка до ЛР8	16
17	Опрацювання лекційного матеріалу. Підготовка до захисту ЛР8. Підготовка до ТК	10
	Разом за семестр:	138

3. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів. Зокрема, лекції проводяться в основному словесними методами, наочними з використанням інформаційних технологій, а також з використанням методів проблемного навчання. Лабораторні заняття проводяться з використанням методів частково-пошукового навчання із використанням інформаційних технологій. Практичні заняття проводяться із використанням пояснювально-ілюстративних методів навчання. Самостійна робота передбачає виконання індивідуальних завдань, при розв'язанні яких застосовуються дослідницькі та частково-пошукові методи навчання.

4. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних, лабораторних та практичних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі заліку. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

Оцінку "відмінно" отримує студент за глибоке і повне опанування змісту навчального

матеріалу, в якому він легко орієнтується, понятійного апарату, за уміння зв'язувати теорію з практикою, вирішувати практичні завдання, висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення. Студент повинен набути практичних навичок із побудови програмних додатків на мові Java із використанням технік об'єктно-орієнтованого програмування.

Оцінка "відмінно" виставляється студенту, який глибоко засвоїв основні принципи розробки крос-платформених додатків із використанням класів, методів та способів збереження, модифікації та відображення даних та вміє їх раціонально застосовувати. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку "добре" отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування в вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку "добре" отримує студент за правильну відповідь з однією суттєвою помилкою.

Оцінки "задовільно" заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок у складанні програм, але допустив неточності. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки "задовільно" заслуговує студент за неповне опанування програмного матеріалу, але отримані знання і набуті практичні навички із розробки програм на основі вивчених алгоритмів.

Оцінка "незадовільно" виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

Оцінка "незадовільно" виставляється студенту за повне незнання і нерозуміння навчального матеріалу або відмову від відповіді і передбачає повторне навчання студента з дисципліни.

Кожний вид роботи оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

Аудиторна робота						Самостійна, індивідуальна робота		Форма семестрового контролю	
ІІІ семестр									
Лабораторні та практичні роботи №:								Тестовий контроль:	
1	2	3	4	5	6	7	8	Т 1-16	залік
ВК:						0,6		0,4	

Примітка: Т – тема дисципліни; ВК – ваговий коефіцієнт;

Для переходу від вітчизняної оцінки до оцінки за шкалою ECTS необхідно знайти середньоарифметичну оцінку за вітчизняною шкалою, помножити її на відповідний ваговий коефіцієнт і, додавши всі складові, отримаємо суму балів, які визначають конкретну оцінку ECTS.

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Інституцій на шкала балів	Інституцій на оцінка	Критерії оцінювання	
A	4,75-5,00	5	Зараховано	Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків.
B	4,25-4,74	4		Добре – повне знання навчального матеріалу з кількома незначними помилками.
C	3,75-4,24	4		Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками.
D	3,25-3,74	3		Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією.
E	3,00-3,24	3		Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00-2,99	2	Незараховано	Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1,99	2		Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни.

Залік виставляється при отриманні студентом з дисципліни від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ECTS – оцінка, що відповідає набраній студентом кількості балів.

При викладанні дисципліни використовуються такі види навчальних занять як лекції, лабораторні роботи, практичні роботи, індивідуальне консультування і керівництво самостійною роботою студента.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: допуск до виконання лабораторної роботи здійснюється на її початку проведенням 10-хвилинної контрольної роботи; засвоєння теоретичного матеріалу тем перевіряється тестовим контролем; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи.

Оцінка, яка виставляється за лабораторну роботу, складається з таких елементів: повнота та якість виконання завдання; знання теоретичного матеріалу з теми; якість оформлення протоколу; вміння студента обґрунтувати прийняті рішення; своєчасний захист лабораторної роботи.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на поточному або наступному після виконання роботи занятті. За несвоєчасний захист лабораторної роботи з набраної студентом суми балів вираховується один бал.

Пропущене з поважної причини лабораторне заняття студент повинен відпрацювати в лабораторіях кафедри у встановлений викладачем термін.

Оцінка, яка виставляється за практичну роботу, складається з таких елементів: якість та повнота виконання завдання; якість оформлення протоколу; своєчасна здача практичної роботи.

Термін захисту практичної роботи вважається своєчасним, якщо студент задав її на наступному після виконання роботи занятті. Практичні роботи, що без поважної причини здані не вчасно, оцінюються оцінкою незадовільно.

Пропущене з поважної причини практичне заняття студент повинен відпрацювати в лабораторіях кафедри у встановлений викладачем термін.

5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ

1. Життєвий цикл програми.
2. Java Virtual Machine.
3. Змінні.
4. Типи даних.
5. Явне приведення типів.
6. Неявне приведення типів.
7. Пріоритет операцій.
8. Методи та їх сигнатури.
9. Класи та об'єкти.
10. Поняття базового класу та його властивості.
11. Поля та методи об'єктів.
12. Ключові принципи ООП.
13. Модифікатори доступу.
14. Методи доступу до приватних полів.
15. Конструктори класів.
16. Абстрактні класи та інтерфейси.
17. Вкладені класи.
18. Клас Object.
19. Масиви та їх методи.
20. Поняття колекцій.
21. Інтерфейс Collection.
22. Колекції.
23. ArrayList, LinkedList.
24. HashSet, Dictionary.
25. Потоки InputStream та OutputStream.
26. Форматований вивід.
27. PrintStream, PrintWriter.
28. Класи для роботи із файлами.
29. FileInputStream, FileOutputStream.
30. Робота із каталогами.
31. Засоби побудови інтерфейсів у Java.
32. Графічні елементи.
33. Компонування елементів.
34. Кнопки та мітки.
35. CheckBox та RadioButton.
36. Події елементів.
37. Технологія JDBC.
38. Підключення до MySQL.
39. Маніпуляції із даними за допомогою Java.
40. Методи executeUpdate та executeQuery.
41. Сокети.
42. Мережеві класи та інтерфейси.
43. Поняття клієнта та сервера у мережевому програмуванні.
44. Способи організації взаємодії між клієнтом та сервером.
45. Клас Thread.
46. Створення та завершення потоку.
47. Синхронізація потоків.
48. Семафори.
49. Обмін даними між потоками.
50. Класи BigInteger та BigDecimal.

6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни забезпечений необхідними навчально методичними розробками в модульному середовищі.

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Core Java(TM), Volume I—Fundamentals, Cay S. Horstmann 2018p – 928p.
2. Head First. Java, Kathy Sierra, Bert Bates 2022p – 688p.
3. Thinking In Java 4th Edition, Bruce Eckel, 2022p – 1168p.
4. DevOps Tools for Java Developers. Best Practices from Source Code to Production Containers, Stephen ChinMelissa McKayIxchel RuizBaruch Sadogursky 2022– 342p.
5. Java Cookbook: Problems and Solutions for Java Developers 4th Edition, Ian F. Darwin, 2020 – 600p.
6. Programming AWS Lambda: Build and Deploy Serverless Applications with Java 1st Edition, John ChapinMike Roberts, 2020 – 224p.
7. Herbert Schildt. Java. The complete Reference. Comprehensive Coverage of the Java Language. Eleventh Edition. McGraw-Hill Education (Publisher), 2019 – 1248p.
8. Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft. Modern Java in Action. Lambdas, streams and functional programming. Manning Publications, 2019 – 592p.

8. ІНФОРМАЦІЙНІ РЕСУРСИ

Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного та семестрового контролю знань).
2. Електронна бібліотека університету.